

Technical Report Machine Learning

Deep Learning Using PyTorch



Disusun Oleh:

Rizky Ramadhani Syam

TK-44-03

1103204086

Program Studi S1 Teknik Komputer

Fakultas Teknik Elektro

Universitas Telkom

2023

1. Deep Learning

Deep learning adalah salah satu subbidang dalam bidang kecerdasan buatan (artificial intelligence) yang berfokus pada pengembangan dan penggunaan algoritma-algoritma yang mirip dengan struktur dan fungsi jaringan saraf manusia. Tujuan utama dari deep learning adalah untuk mengembangkan sistem yang mampu belajar secara mandiri melalui pengalaman dan data, serta mampu melakukan tugas-tugas kompleks dengan kinerja yang semakin meningkat seiring waktu.

Deep learning menggunakan jaringan saraf tiruan yang terdiri dari banyak lapisan (layers) yang saling terhubung. Setiap lapisan ini terdiri dari unit-unit pemrosesan (neuron) yang menghitung dan mentransformasikan input menjadi output. Lapisan-lapisan tersebut membentuk hierarki pemrosesan informasi, di mana lapisan-lapisan awal belajar fitur-fitur sederhana dari data mentah, sedangkan lapisan-lapisan yang lebih dalam belajar fitur-fitur yang lebih kompleks.

Proses pembelajaran dalam deep learning dilakukan melalui dua tahap utama, yaitu tahap pelatihan (training) dan tahap pengujian (inference). Pada tahap pelatihan, deep learning menggunakan algoritma pembelajaran yang disebut backpropagation, yang memungkinkan jaringan saraf untuk menyesuaikan bobot-bobot (weights) dan bias-bias yang terkait dengan setiap neuron dalam jaringan. Pelatihan dilakukan dengan mempersembahkan sejumlah besar data latihan (training data) yang berisi contoh-contoh yang sudah diketahui keluarannya, sehingga jaringan saraf dapat belajar menyesuaikan diri dan menghasilkan keluaran yang benar. Setelah tahap pelatihan selesai, deep learning dapat digunakan untuk melakukan prediksi atau klasifikasi terhadap data baru pada tahap pengujian. Jaringan saraf yang telah terlatih akan menerima input baru dan menghasilkan output berdasarkan pola-pola yang telah dipelajari selama tahap pelatihan. Keunggulan deep learning terletak pada kemampuannya untuk mengenali pola-pola kompleks dan mampu belajar dari data yang tidak terstruktur.

Deep learning telah diterapkan dalam berbagai bidang, termasuk pengenalan wajah, deteksi objek, pengenalan suara, analisis teks, pengolahan bahasa alami, dan banyak lagi. Keberhasilan deep learning dalam beberapa tahun terakhir dapat dikaitkan dengan beberapa faktor, seperti kemajuan dalam komputasi yang memungkinkan pelatihan jaringan saraf yang lebih besar dan kompleks, serta ketersediaan data yang lebih besar untuk pelatihan.

Namun, deep learning juga memiliki beberapa tantangan. Salah satu tantangan utama adalah interpretabilitas model. Karena deep learning menggunakan jaringan saraf yang kompleks, sulit untuk memahami alasan di balik setiap keputusan yang diambil oleh model. Selain itu, deep learning juga membutuhkan sumber daya komputasi yang besar dan waktu pelatihan yang lama, terutama untuk jaringan yang sangat dalam dan data yang besar.

Secara keseluruhan, deep learning telah mengubah lanskap kecerdasan buatan dengan memberikan kemampuan yang luar biasa dalam memahami dan memproses data kompleks. Dengan terus berkembangnya teknologi dan penelitian di bidang ini, deep learning diharapkan dapat memberikan kontribusi yang signifikan dalam berbagai aspek kehidupan manusia di masa depan.

2. PyTorch

PyTorch adalah sebuah framework atau kerangka kerja (framework) sumber terbuka yang populer dalam bidang kecerdasan buatan (artificial intelligence) dan pengolahan data. Dikembangkan oleh Facebook AI Research (FAIR), PyTorch menawarkan alat dan fungsi yang kuat untuk membangun, melatih, dan menerapkan model jaringan saraf tiruan (neural networks) dalam berbagai aplikasi.

Salah satu keunggulan utama PyTorch adalah pendekatan yang intuitif dan fleksibel dalam pemrograman yang memungkinkan para peneliti dan praktisi di bidang kecerdasan buatan untuk dengan mudah mengembangkan model mereka. PyTorch menggunakan pendekatan yang disebut "Define-by-Run", di mana pengguna dapat secara dinamis mendefinisikan dan mengubah graf komputasi saat eksekusi program berlangsung. Berikut adalah beberapa fitur utama dari PyTorch:

1. **Dinamika Komputasi:** PyTorch memungkinkan pengguna untuk melakukan operasi matematika dan komputasi tensor dengan cara yang mirip dengan numpy array. Tensor adalah struktur data utama dalam PyTorch dan memungkinkan pengguna untuk melakukan operasi seperti penjumlahan, perkalian, dan fungsi matematika lainnya.
2. **AutoGrad:** PyTorch menyediakan fungsionalitas otomatis diferensiasi (automatic differentiation) melalui modul AutoGrad. Dengan AutoGrad,

PyTorch secara otomatis menghitung gradien (gradient) dari suatu fungsi terhadap parameter yang diperlukan dalam proses pelatihan model jaringan saraf. Ini memudahkan pelatihan model dan penggunaan algoritma pembelajaran berbasis gradien seperti backpropagation.

3. **Modul Jaringan Saraf:** PyTorch menyediakan berbagai modul yang dapat digunakan untuk membangun arsitektur jaringan saraf. Pengguna dapat dengan mudah mendefinisikan lapisan-lapisan, fungsi aktivasi, dan modul lainnya untuk membangun model yang kompleks. PyTorch juga menyediakan banyak modul jaringan saraf yang telah dibangun sebelumnya seperti Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), dan lain-lain.
4. **Penanganan GPU:** PyTorch memiliki dukungan yang kuat untuk komputasi GPU. Hal ini memungkinkan para pengguna untuk dengan mudah memanfaatkan kecepatan dan kekuatan komputasi paralel yang ditawarkan oleh GPU untuk melatih model mereka dengan cepat.
5. **Alat dan Utilitas:** PyTorch menyediakan berbagai alat dan utilitas untuk membantu pengguna dalam membangun, melatih, dan mengevaluasi model mereka. Ini termasuk fungsi untuk memuat dan mengolah data, alat visualisasi, fungsi pengukuran kinerja, dan banyak lagi.
6. **Integrasi dengan Ekosistem Python:** PyTorch diintegrasikan dengan baik dengan ekosistem Python yang luas. Pengguna dapat dengan mudah menggunakan pustaka Python lainnya seperti NumPy, SciPy, dan Pandas bersamaan dengan PyTorch untuk analisis data yang efisien.
7. **Komunitas yang Aktif:** PyTorch memiliki komunitas yang aktif dan berkembang pesat. Komunitas ini menyediakan sumber daya, tutorial, dan dukungan yang kaya bagi pengguna PyTorch. Juga, terdapat banyak model jaringan saraf yang telah dilatih sebelumnya yang tersedia secara gratis untuk digunakan atau diubah sesuai kebutuhan.

PyTorch telah digunakan secara luas dalam berbagai bidang, termasuk pengenalan gambar, pemrosesan bahasa alami, visi komputer, pengolahan suara, dan banyak lagi. Dengan kemampuannya yang kuat dan pendekatan yang intuitif, PyTorch terus menjadi salah satu pilihan utama bagi para peneliti dan praktisi di bidang kecerdasan buatan.

3. Arsitektur Pytorch

Di dalam PyTorch terdapat struktur data `torch.Tensor` yang merupakan representasi dari array multidimensi dan mendukung perhitungan numerik yang efisien. Tensor dapat dibuat, dimanipulasi, dan dioperasikan menggunakan berbagai operasi matematika dan logika. PyTorch menggunakan GPU untuk melakukan operasi tersebut secara paralel, memberikan peningkatan kecepatan yang signifikan.

PyTorch juga menyediakan abstraksi yang kuat bernama `torch.nn`, yang memungkinkan pengguna untuk mendefinisikan model jaringan saraf menggunakan komponen modular yang disebut "modul". Modul mengkapsulasi arsitektur dan parameter yang dapat dipelajari dari jaringan saraf. Dengan mewarisi kelas `torch.nn.Module`, pengembang dapat membuat arsitektur jaringan saraf kustom, mendefinisikan forward pass, dan menentukan bagaimana parameter jaringan diperbarui selama pelatihan. Proses pelatihan dalam PyTorch umumnya melibatkan langkah-langkah berikut:

- Mendefinisikan arsitektur jaringan saraf dengan membuat modul kustom.
- Menyiapkan data dan membuat data loader untuk pengelompokan data yang efisien dan data augmentation.
- Menentukan fungsi kerugian yang mengukur perbedaan antara prediksi jaringan dengan kebenaran sejati.
- Mengonfigurasi optimizer, seperti stochastic gradient descent (SGD) atau Adam, untuk memperbarui parameter model berdasarkan gradien yang dihitung.
- Melakukan iterasi pada data pelatihan, melakukan forward dan backward pass, dan memperbarui parameter model.
- Mengevaluasi model yang telah dilatih pada set validasi atau set pengujian terpisah untuk menilai performanya.

4. Penggunaan PyTorch Pada Kode

a. Tensor Basic Tensor

merupakan struktur data dasar dalam PyTorch yang digunakan untuk menyimpan dan mengolah data numerik.

b. Autograd Autograd

adalah fitur dalam PyTorch yang mengotomatisasi perhitungan gradien. Hal ini memungkinkan kita untuk menghitung gradien dari suatu fungsi terhadap parameter-parameter yang diperlukan.

c. Backpropagation Backpropagation

adalah algoritma yang digunakan untuk menghitung gradien dari fungsi loss terhadap parameter-parameter model dalam jaringan saraf. Dalam PyTorch, backpropagation dilakukan secara otomatis melalui autograd.

d. Gradient Descent With Autograd and Backpropagation Gradient Descent

adalah algoritma optimisasi yang digunakan untuk meminimalkan fungsi loss dengan memperbarui parameter-model berdasarkan gradien. Dalam PyTorch, kita dapat menggunakan autograd dan backpropagation untuk menghitung gradien dan melakukan optimisasi Gradient Descent.

e. Training Pipeline: Model, Loss, and Optimizer

Dalam pipeline pelatihan, membutuhkan model, fungsi loss, dan optimizer.

f. Linear Regression Linear Regression

adalah metode statistik untuk memodelkan hubungan linier antara variabel input dan output.

g. Logistic Regression Logistic Regression

adalah metode yang digunakan untuk klasifikasi biner.

h. Dataset and DataLoader Dataset dan DataLoader

adalah komponen yang digunakan untuk memuat dan mengelompokkan data dalam pelatihan model.

i. Dataset Transforms

Transformasi dataset digunakan untuk melakukan transformasi pada data sebelum dimasukkan ke dalam model.

j. Softmax And Cross Entropy Softmax dan Cross Entropy

adalah fungsi yang umum digunakan dalam klasifikasi multikelas.

k. Activation Functions

Fungsi aktivasi digunakan untuk memperkenalkan sifat non-linear ke dalam model jaringan saraf.

l. Feed-Forward Neural Net Feed-Forward Neural Network

adalah jenis jaringan saraf yang informasinya mengalir hanya ke arah maju, dari input menuju output.

m. Convolutional Neural Net (CNN) Convolutional Neural Network

adalah jenis jaringan saraf yang digunakan khusus untuk tugas-tugas pengolahan citra.

n. Transfer Learning

Transfer Learning adalah teknik yang memanfaatkan pengetahuan yang telah dipelajari dari model sebelumnya untuk mempercepat pelatihan model pada tugas baru.

o. Tensorboard Tensorboard

adalah alat visualisasi yang digunakan untuk memonitor dan menganalisis pelatihan model.

p. Save and Load Models

Untuk menyimpan dan memuat model yang telah dilatih, PyTorch menyediakan metode untuk menyimpan dan memuat parameter-parameter model.

5. Kesimpulan

Dalam laporan ini, telah dibahas mengenai Machine Learning dan PyTorch dengan cara yang sederhana. Machine Learning merupakan bidang ilmu komputer yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau melakukan tindakan berdasarkan pola yang ditemukan dalam data tersebut. PyTorch adalah salah satu alat yang digunakan dalam Machine Learning, yang membantu dalam pengembangan dan pelatihan model untuk tugas-tugas seperti pengenalan gambar, klasifikasi data, dan prediksi.

PyTorch memiliki beberapa fitur yang membuatnya populer di kalangan pengembang Machine Learning. Misalnya, PyTorch menggunakan grafik komputasi dinamis yang memudahkan dalam pembuatan dan modifikasi model. Ini memungkinkan eksperimen yang mudah dan mempermudah proses debugging. Selain itu, PyTorch juga mendukung penggunaan GPU untuk mempercepat perhitungan dan memiliki ekosistem aktif dengan banyak pustaka dan sumber daya yang dapat digunakan.

Dalam PyTorch, digunakan struktur data yang disebut tensor untuk menyimpan dan mengolah data numerik. Autograd adalah fitur yang membantu dalam perhitungan gradien dengan mudah, yang penting dalam pelatihan model. Backpropagation adalah algoritma yang digunakan untuk menghitung gradien secara otomatis. Dengan bantuan algoritma optimisasi seperti Gradient Descent, parameter-model dapat diperbarui untuk meminimalkan kesalahan prediksi.

Juga dipelajari beberapa konsep dan komponen penting dalam PyTorch seperti Linear Regression (untuk memodelkan hubungan linier antara variabel input dan output), Logistic Regression (untuk klasifikasi biner), Convolutional Neural Network (CNN) (untuk pengolahan citra), dan Transfer Learning (menggunakan pengetahuan dari model sebelumnya untuk mempercepat pelatihan model baru).

Selain itu, PyTorch juga menyediakan berbagai utilitas yang membantu dalam memuat dan mengelompokkan data, melakukan transformasi pada data, dan menyimpan serta memuat model yang telah dilatih. Semua ini memudahkan dalam pengembangan aplikasi Machine Learning dengan PyTorch.

Secara keseluruhan, PyTorch adalah alat yang populer dan efektif untuk pengembangan aplikasi Machine Learning. Dengan fitur-fitur yang kuat, fleksibilitas, dan dukungan yang luas, PyTorch telah digunakan dalam berbagai bidang seperti pengenalan gambar, pemrosesan bahasa alami, dan visi komputer.